Sapphire Sounds

Sapphire Group

Prototype Design Presentation

7AUG2025

Table of Contents

Meet the Team

Priscilla Cook

Wes Brown

Brian Stiles

Zach Boudreaux

Shawn Maybush

Zachery Hendrickson

Alisa Smanpongse

Yahya Mohamed

Christiney Ponton

Elevator Pitch

PROBLEM	Noise complaints are a common issue in shared buildings. These complaints can often lead to disputes, unfair accusations, and unresolved tension between tenants.
SOLUTION	A software-enabled noise detector that will be paired with a mobile or web app to monitor decibel levels, without recording audio. When noise levels are exceeded for an extended period the system will log the event and generate a report for property managers.
KEY FEATURES	Decibel Tracking Real time alerts Noise history dashboard Auto generated incident reports
BENEFITS	Tenants enjoy peace and quiet, protection from false complaints and property managers receive unbiased data to resolve disputes.
OUTCOME	A more peaceful, quiet, and respectful living environment, driven by transparent data and ethical technology.

Problem Statement

In shared living environments, current noise complaint processes rely heavily on subjective reports, leading to unresolved disputes, false accusations, and tenant dissatisfaction. There is no efficient, unbiased, or privacy-conscious method for tracking and verifying noise disturbances over time.

Problem Characteristics

High prevalence of renters

- About **36%** of U.S. households are renter-occupied, highlighting the widespread relevance of noise disputes in shared living environments (Parker et al.).

Frequent noise complaints

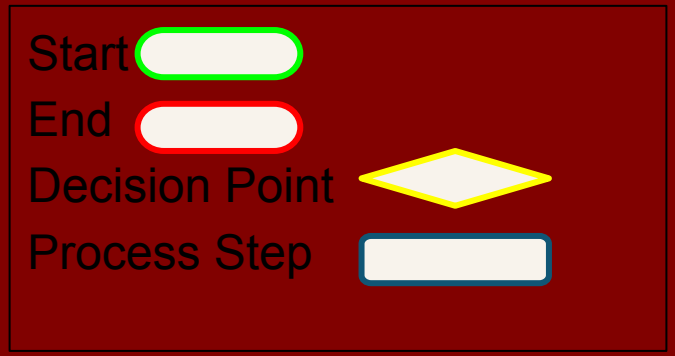
- Nearly **40%** of people have lodged noise complaints, yet only **20%** feel their issues were resolved satisfactorily (Bronzaft and Nadler).

Subjective and unsubstantiated claims

- With **80%** of complainants not having their noise issues addressed—even after formal complaints—there is a clear need for objective evidence (Bronzaft and Nadler).

Inefficiencies in landlord/tenant resolution

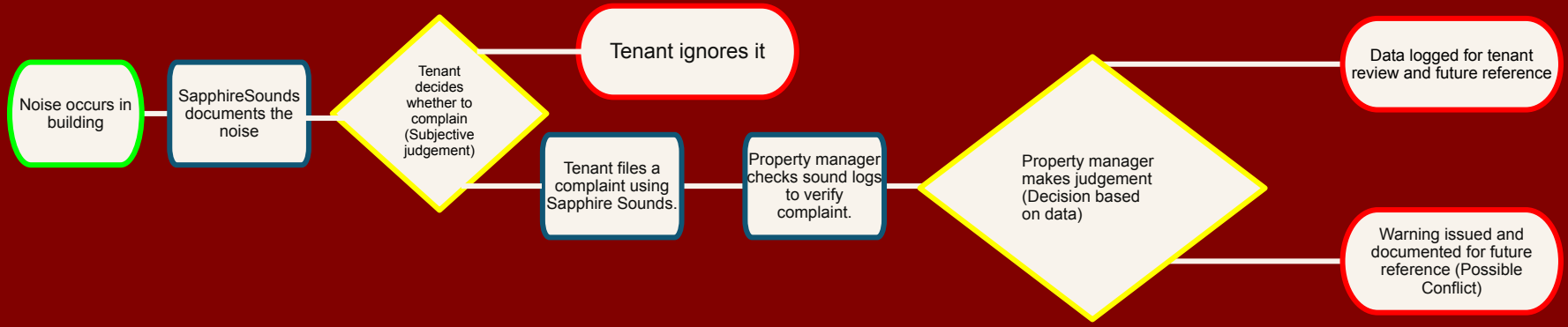
- Without reliable documentation, property managers struggle to handle noise reports fairly, leading to tenant frustration and strained relationships.

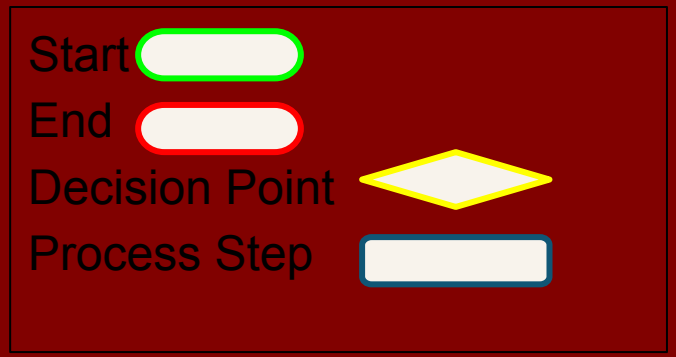


Current Process Flow

Solution Statement

Sapphire Sounds is a mobile and web application that connects to a small noise sensor. The sensor tracks how loud sounds are but does not record any audio. If noise levels remain too high for too long, Sapphire Sounds generates a report with the time and noise level. These reports help property managers resolve complaints fairly. Tenants can view their own noise history, receive alerts when they’re too loud, and earn rewards for maintaining quiet. Sapphire Sounds promotes peaceful shared living by providing clear, unbiased data instead of relying on personal opinions.





Solution Process Flow

What it will do

Objectively monitor noise levels

- Use sensors to to detect and log persistent noise violations.
- Capture sound intensity in decibels without recording audio directly, respecting the privacy of tenants.

Provide time-stamped noise report

- Generate logs that show when noise events occur, how long they last, and how loud they are.
- Identify patterns of noise violations.

Support conflict resolution

- Share noise data with affected parties (roommates, tenants, landlords, or property managers) to support fair, evidence-based discussions.
- Use objective data to de-escalate disputes by replacing subjective claims with measurable evidence.

Promote healthier shared living arrangements

- Help maintain a peaceful, respectful environment by discouraging chronic noise violations.

What it will not do

Focus on Isolated, One-Time Noise Events

- Isolated incidents typically do not indicate disruptive behavior. This may lead to unnecessary reports.

Interpret Meaning or Intent Behind Noise

- The system will not analyze the type or intent of sounds, such as distinguishing between arguments, music, or social gatherings. This would require capturing audio, which raises privacy concerns and contradicts Sapphire Sound Monitor’s privacy-first design.
- The system is focused solely on measuring decibel levels. It will not distinguish between different noise sources such as pets, appliances, or people.

Real-Time Noise Enforcement

- Handling the noise conflict will ultimately be left up to property management. Sapphire Sound Monitor is designed to provide objective evidence to support fair decision-making.

Tenants Earn Rewards

Sapphire Sounds encourages tenants to maintain low noise levels by allowing property managers to offer incentives such as discounts, recognition badges, or community perks.

The Rewards System

- **The rewards system will provide property managers the ability to create rewards for tenants who maintain respectful noise levels for a consecutive amount of time.**
- **Property managers will designate the amount of time before a reward is earned.**
  - For instance, a property manager can create a reward for maintaining respectful noise levels for the entire month.
  - The property manager could also have an even better reward to be earned for six consecutive months of no alerts.
  - Time units will be in either days or months.
- **The rewards themselves will be determined and provided by the property manager.**
  - This could be anything from rent rebates to movie tickets.
- **Tracking of whether a reward has been claimed or not will be provided by Sapphire Sounds and will be updated by the property manager.**

Log Access Feature

Tenants will have access to a dashboard where they can view:
- Historical noise events
- Time-stamped decibel logs
- Their compliance record over time

This transparency helps tenants self-regulate and resolve complaints proactively.

Competition Matrix

	Sapphire Sounds	MINUT	RentEye	Traditional Complaint
Privacy (no audio recordings)	✅	✅	✅	✅
Notifications to owner		✅	✅	
Notifications to tenants	✅	✅	✅	
Historical Data	✅	✅	✅	✅
Tamper Detection			✅	
Remote Monitoring	✅	✅	✅	
Tenants Initiate Complaints through the app	✅			
Focused on Preventing/Resolving Noise Complaints	✅			
Tenants Earn Rewards	✅			

Minut. Minut.com. https://www.minut.com
RentEye. RentEye.com. https://www.renteye.com

Development Tools

- **IDE** - VSCode
- **Version Control** - Git with GitHub
- **CI/CD** - GitHub Actions & Workflows
- **Backend Language** - JavaScript (Node.js)
- **Frontend Languages** - JavaScript (React.js), HTML, CSS
- **Database Management System -** PostgreSQL
- **Testing Framework** - Jest, Postman
- **Documentation Tool** - JSDoc
- **Sensor Control Language** - C++
- **Sensor Communication RESTFul API** - Express.js
- **Sensor Control Driver** - i2c-tools

Required Libraries

- **Backend**
  - **node-postgres** - postgres client
  - **Danfo.js** - Data structures for manipulating and processing data. This is the node equivalent to pandas.
  - **TensorFlow.js** - Array manipulation. Seamless integration with Danfo.js
  - **NumJs** - alternative to TensorFlow.js. Lightweight but not integrated with Danfo.js.
- **Frontend**
  - **React.js** - framework
  - **react-router-dom** - provides efficient routing within React web applications.
  - **React-Bootstrap** - component library for Bootstrap style components.
- **Other**
  - **Jest** - unit testing
  - **JSDoc** - documentation

Major Functional Components

Software Components

- **Firebase** - For hosting our React.JS frontend.
- **Render.com** - For hosting our Node.JS backend.
- **Clever Cloud** - For hosting our PostgreSQL databases.
- **GitHub** - Beyond repository management, also an origin to deploy to a hosting service.
- **Docker & Docker Compose** - For localized hosting during development.
- **Sensor Driver** - i2c-tools Linux Package

Sensor Components

- **Raspberry Pi Zero 2 W** - Controller to process sensor readings and send them to the application server.
- **I2C Decibel Sound Level Meter Module** - This sensor will connect to the Raspberry Pi via the I2C Bus (created by PCB Artist).
- **Infineon CYW43439** - WiFi connectivity module built into the Raspberry Pi Zero 2 W.

Off The Shelf Software

- **Node.JS** - Backend
- **React.JS** - Frontend
- **PostgreSQL** - RDBMS
- **Decibel Meter Driver**

Off The Shelf Hardware

- **Raspberry Pi Zero 2 W** - Sensor Controller
- **I2C Decibel Sound Level Meter Module** - Acoustic Sensor
- **Infineon CYW43439** - Connectivity Module

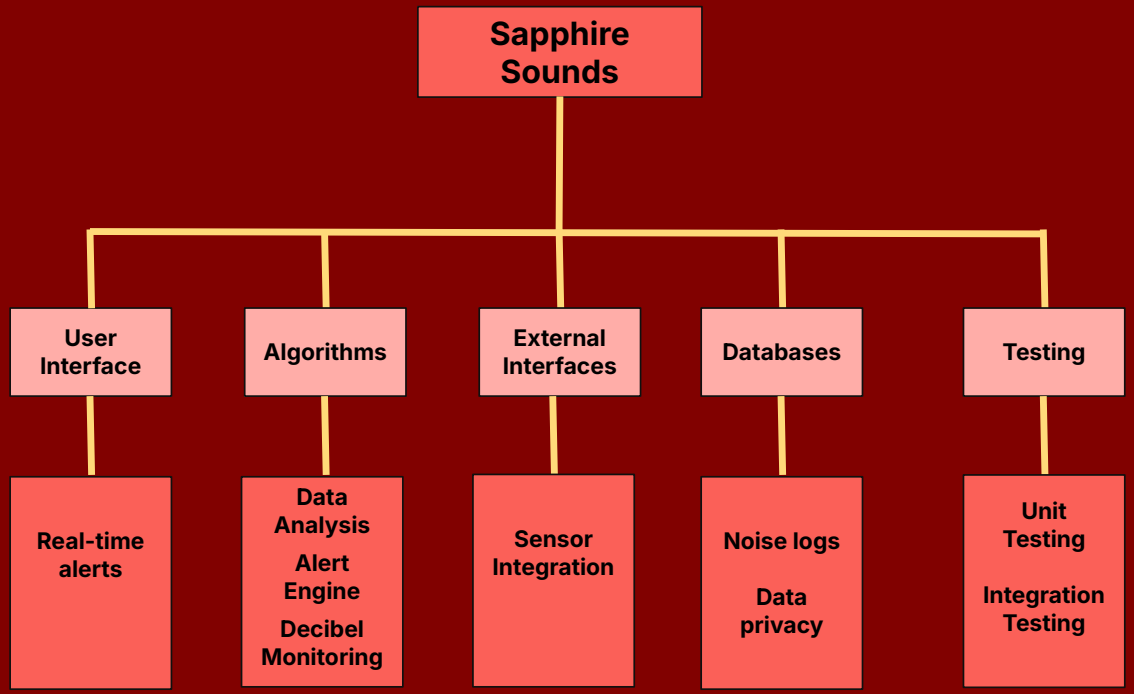
What We Will Build

- Data Analysis Engine
- Alert Engine
- Decibel Monitoring
- Decibel Meter Communication

Major Functional Components Diagram

What we will build

- Data Analysis Engine
  - Anomaly detection
  - Noise profiling by time of day or day of week
  - Noise comparisons across different time ranges
- Alert Engine
  - Real-time alerts to tenants, landlords, or property managers
  - Configurable escalation logic
  - Summarized reports and dashboards
- Decibel Monitoring Logic
  - Real-time decibel level collection and timestamping
  - Short and long-term noise averaging
  - Identification of noise spikes or sustained high volumes that exceed predefined thresholds
  - Decibel Meter to server communication

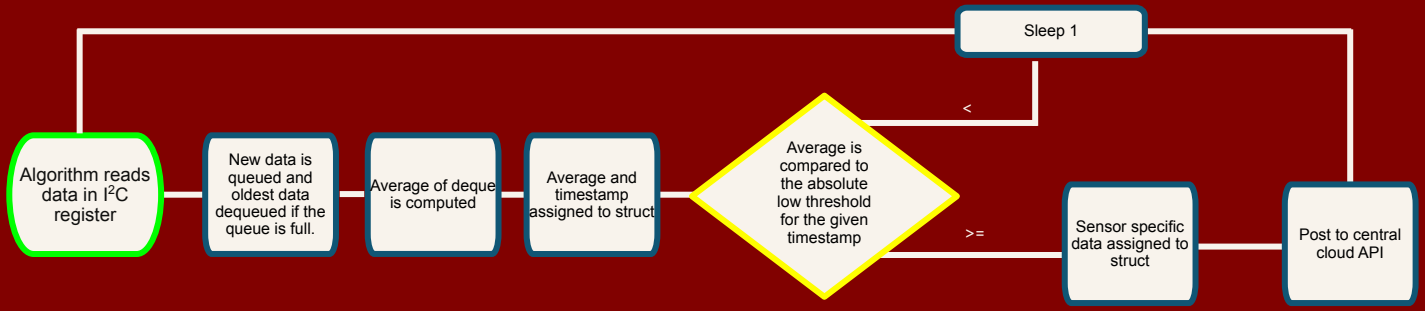


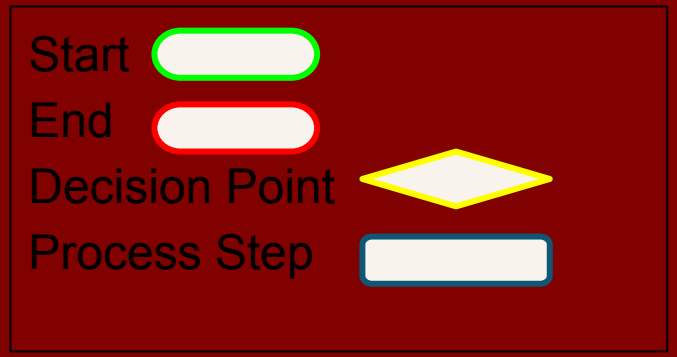
Work Breakdown Structure

Decibel Monitor Algorithm

The Decibel Monitor logic implements processing to detect and report noise events.

- **I2C Communication:** The Raspberry Pi communicates with the dB sensor via the I2C protocol.
- **Fixed-Window Averaging using a Deque:**
  - Sensor readings are stored in a double-ended queue.
  - The size of the deque is a configurable setting (averageWindowSeconds) downloaded from the server.
  - With each new 1-second reading, the oldest reading is removed from the front of the deque, and the new reading is added to the back.
  - A running average is then calculated from all the readings currently within the deque. This process ensures that only sustained noise levels are considered.
- **Threshold Evaluation:** The calculated fixed-window average is compared against a low absolute threshold (thresholdLowAbsolute) downloaded from the server (defaulting to 65 dB).
- **Event-Driven Reporting:** When a reportable event is detected, the data is sent to the central cloud API along with relevant info for the specific sensor.





Decibel Monitor Process Flow

Alert Engine Algorithm

The Alert Engine defines how users receive notifications of noise events.

- **Receive Report:** The report received from the monitoring system begins to process to notify users.
- **Message Preparation:** A middleware for the backend and databases will prepare a message of the event with pertinent details and identify the users that will receive the communication.
- **Dispatch Notification:** A middleware for the backend and databases will deliver an email (SMTP) and/or SMS message to the appropriate users with the event details outlined above.
- **Logging:** After successfully dispatching the alert notification to users, a report will be logged into a database for users to follow up with in the app.

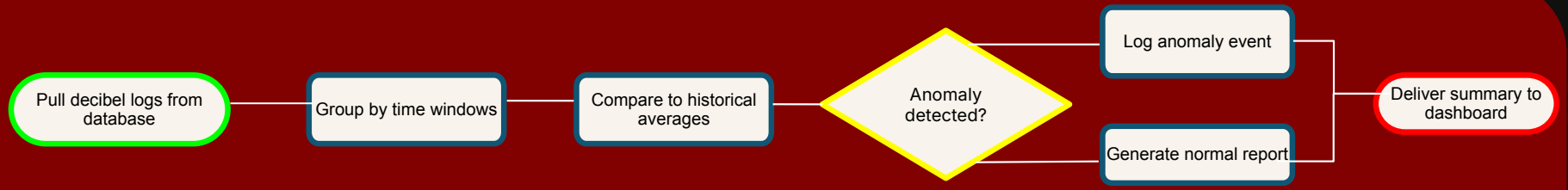
Start
End
Decision Point
Process Step

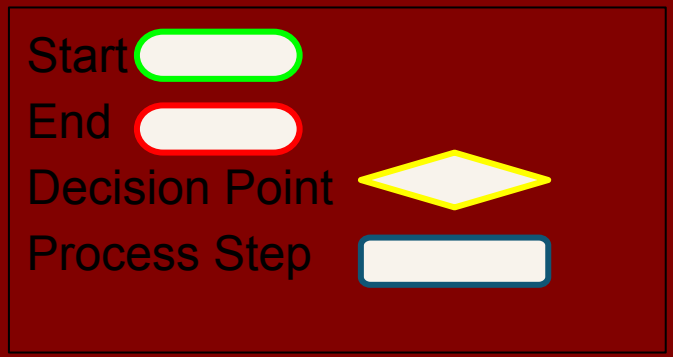
Alert Engine Process Flow

Data Analysis Engine

The Data Analysis Engine processes noise logs to detect trends, flag anomalies, and generate summary reports.

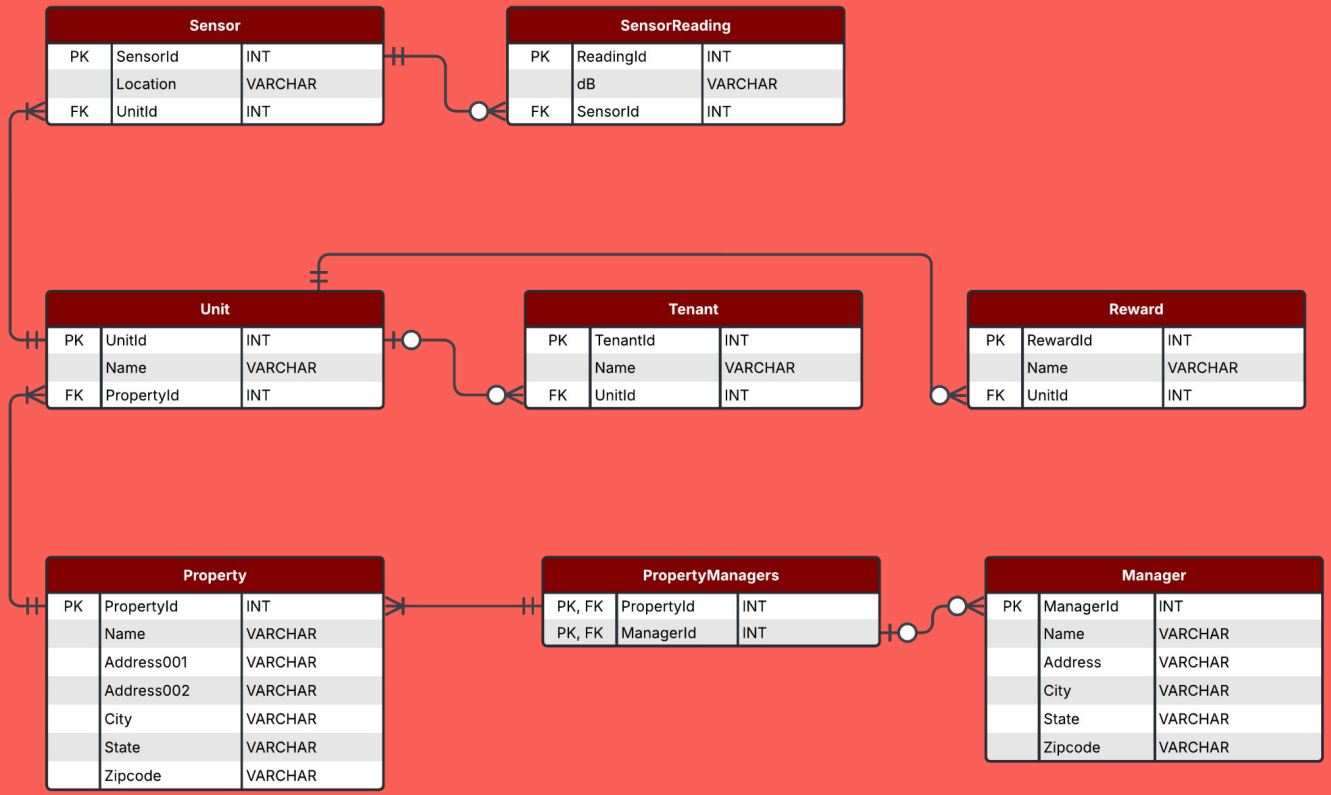
- **Retrieve Logs**: The engine pulls historical decibel readings and noise event data from the database based on selected units,sensors, and timeframes.
- **Time-Based Groupings**: Events are grouped by hour, day, and week to build patterns such as “frequent late-night violations” or “weekend spikes.”
- **Trending Comparison**: Compare recent noise activity to historical baselines to detect trends in increasing frequency, duration, or decibel level of events.
- **Anomaly Detection**: Significant deviations such as sudden spikes or violations during quiet hours are flagged for further review by the system or property managers.
- **Summary Generation**: Periodic reports are generated showing total violations, common violation times, and changes over time.
- **Output Delivery**: Results are forwarded to
    - The dashboard for tenants to self-monitor compliance
    - The property manager interface to assist in data backed conflict resolution
    - The reward system to determine if tenants qualify for perks





Data Analysis Engine Process Flow

Database Schema



Real World Product vs Prototype

Real World

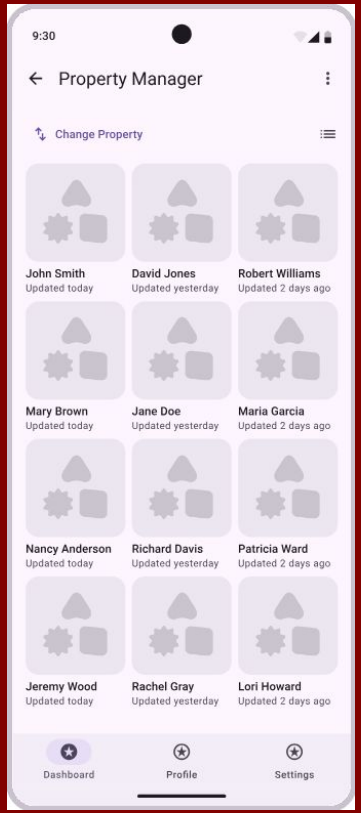
- **Decibel Monitoring** - The sensors will be installed into a physical device and their data will be be recorded from real world events.
- **Alert Engine** - Will trigger alerts using sensors and Wi-Fi connectivity module. Then will send a notification to the landlord's email.
- **Data Analysis Engine** - Will analyze historical sound levels and identify any patterns.
- **Decibel Meter Communication** - Will transmit data in real-time using a cloud server and will also connect to the app.

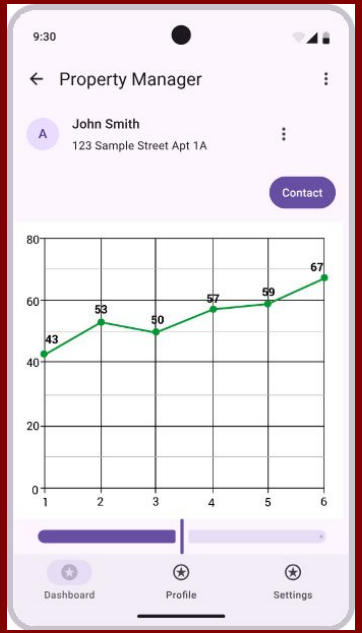
VS

Prototype

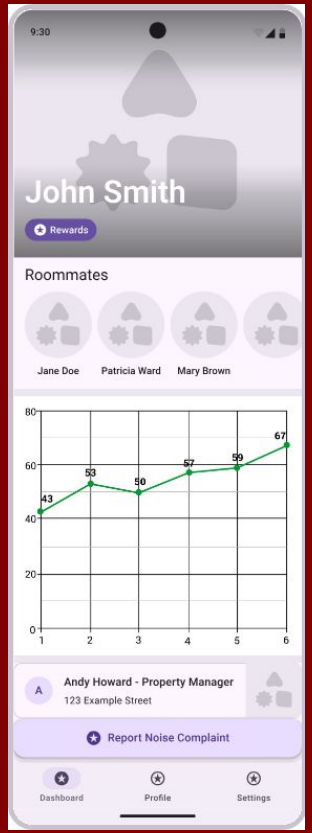
- **Decibel Monitoring** - Will lack sensors and their data will be simulated.
- **Alert Engine** - Will simulate alerts with simple logic.
- **Data Analysis Engine** - Will be limited to a local host and how a detection will be analyzed will use simple rules and logic.
- **Decibel Meter Communication** - The input data will be simulated and will lack utilizing any hardware.

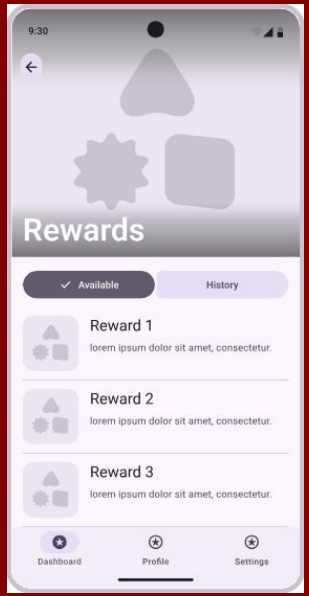
GUI Mockup: Property Manager Interface



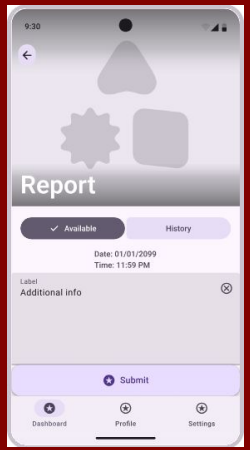


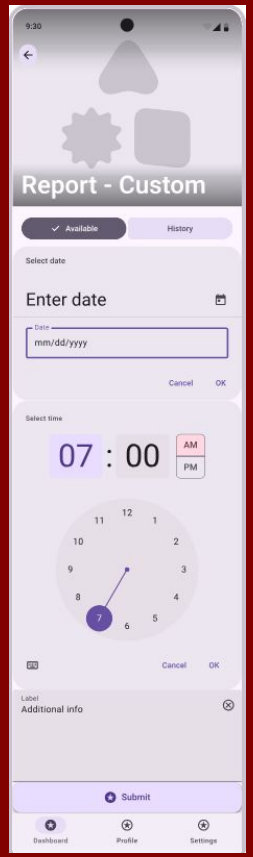
GUI Mockup: Tenant Interface





GUI Mockup: Reporting Interface





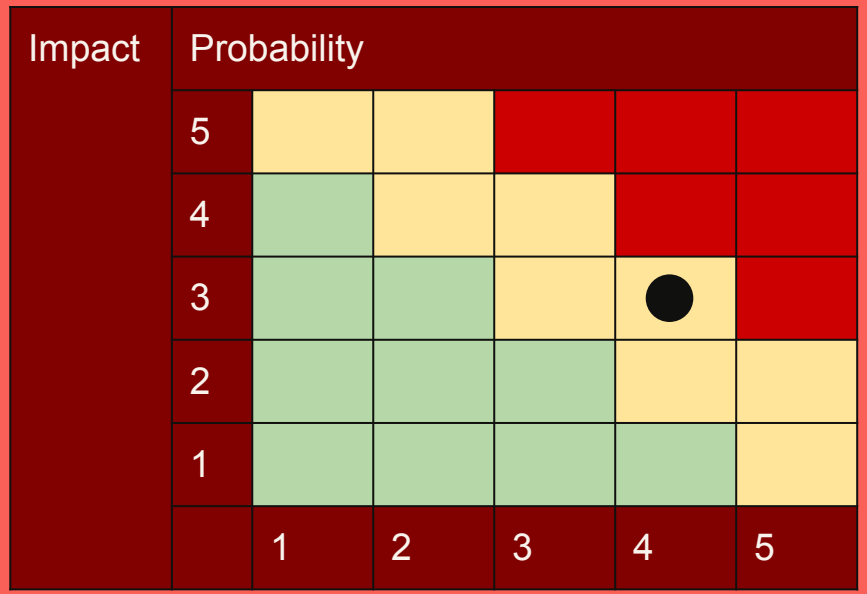
GUI Mockup: Admin Interface

Risks

- Customer & End User
- Technical
- Security
- Legal

Customers & End User Risk

Moderate Risk 🚩



Risk: Hesitation to use the Sapphire Sound app or sensor device

Description: Tenants may feel the system is unfair or intrusive, or they may distrust Sapphire Sound altogether, which could result in low adoption and limited participation.

Probability: 4 (quite likely, due to privacy concerns and low initial trust)

Impact: 3 (may prevent fair and consistent noise enforcement across the community)

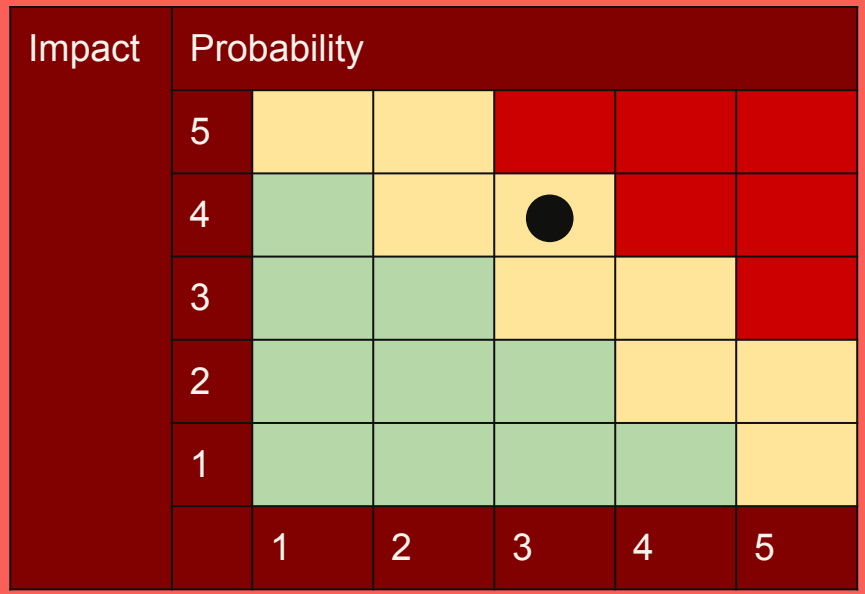
Mitigation:

- Clearly communicate privacy safeguards (no audio recording, only decibel levels)
- Add tenant-friendly features like real-time alerts and a rewards
- Engage community leaders or property managers to champion the system.

Expected Improvement: Probability decreases to 2; impact decreases to 2.

Technical Risk

Moderate Risk 🚩



Risk: Sensor device inaccuracies or failures (false positives/negatives)

Description: Noise sensors may malfunction or misinterpret sounds, leading to incorrect reports or missed noise events. This could undermine trust in the system.

Probability: 3 (moderate, depending on device quality and environment)

Impact: 4 (false reports could cause unfair complaints, missed events reduce system credibility)

Mitigation:

- Use high-quality calibrated sensors with reliable threshold algorithms to effectively disregard irrelevant sounds.
- Include a manual review or appeal process to dispute or clarify flagged noise events
- Consider adaptive tuning for different environments

Expected Improvement: Probability decreases to 1; impact decreases to 2.

Security Risk

High Impact Risk 🚩

Impact	Probability				
5					●
4					
3					
2					
1					
	1	2	3	4	5

Risk: Data breach and unauthorized access

Description: The account connected to the noise sensor could potentially have the user's private information compromised, such as their email, name, and password.

Probability: 5 (High, if the proper security measures are not taken beforehand).

Impact: 5 (Leaked data can lead to identity theft, bank accounts being targeted, personalized targeted scams, and loss of public trust).

Mitigation:

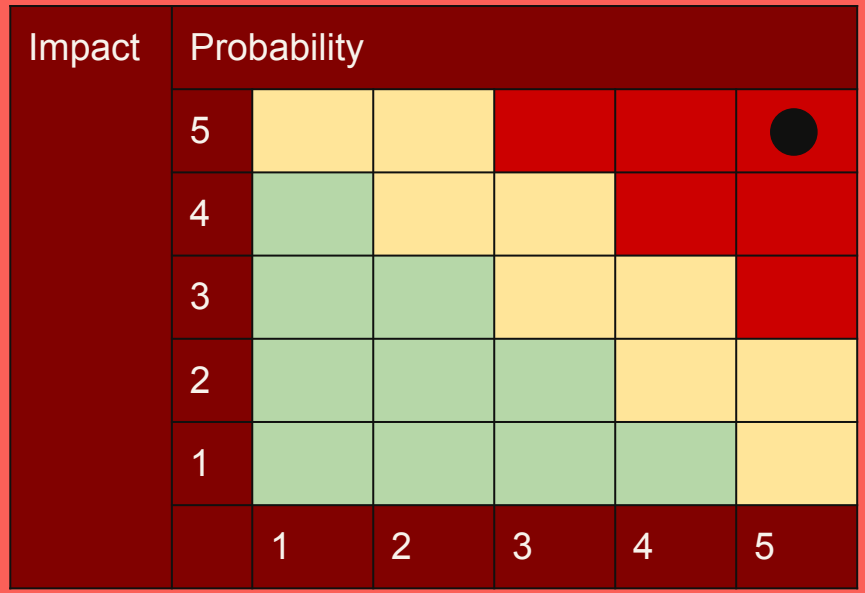
- Device encryption
- Strong Passwords
- 2 Factor Authentication
- Software Updates

Expected Improvement: Probability decreased to 1, but due to the severity of the impact, it will remain at a high level.

Legal Risk

High Impact Risk 🚩

Low | Medium | High



Risk: Privacy violations and data protection laws

Description: The sound device's acoustic sensor can possibly record private conversations and risk breaking data protection laws. If accounts or devices are compromised then the creators of the device can face severe legal consequences.

Probability: 5 (if proper security precautions and consent agreements are not met.)

Impact: 5 (Legal fines and possible incarceration, loss of public trust, lawsuits, devices being banned / recalled)

Mitigation:

- Get user consent through a legally binding contract and disclose the devices usage and functions to all users.
- Make sure any built-in microphones can **only** record decibel levels.
- Have records of compliance for data protection laws.

Expected Improvement: Probability decreased to 1 and impact decreased to 3 (medium).

Risk Conclusion

Most critical risk: Legal (Privacy violations)

- Mitigations in place reduce technical, customer, and security risks significantly
- Continued user transparency, consent, and legal compliance are key to success

Conclusion

Problem: Noise complaints are a common issue in shared buildings. These complaints can often lead to disputes, unfair accusations, and unresolved tension between tenants.

Solution: A software-enabled noise detector that will be paired with a mobile or web app to monitor decibel levels, without recording audio. When noise levels are exceeded for an extended period the system will log the event and generate a report for property managers.

Outcome: A more peaceful, quiet, and respectful living environment, driven by transparent data and ethical technology.

Logo

References

Bronzaft, Arline L., and Richard Nadler. "Community Complaints." *NoiseAwareness.org*, Noise Pollution Clearinghouse, noiseawareness.org/info-center/community-complaints-bronzaft-nadler/. Accessed 1 July 2025.

"Handling Noise Complaints from Tenants." *LawInfo*, www.lawinfo.com/resources/landlord-tenant/handling-noise-complaints-from-tenants.html. Accessed 1 July 2025.

Minut. "Minut." *Minut.com*, n.d., www.minut.com/.

Parker, Kim, and Rachel Minkin. "As National Eviction Ban Expires, a Look at Who Rents and Who Owns in the U.S." *Pew Research Center*, 2 Aug. 2021, www.pewresearch.org/short-reads/2021/08/02/as-national-eviction-ban-expires-a-look-at-who-rents-and-who-owns-in-the-u-s.

RentEye. "RentEye." *RentEye.com*, n.d., www.renteye.com/.

Appendix

User Roles and Stories

User Role: Property Manager

- As a Property Manager, I need to be able to set up a new property.
- As a Property Manager, I need to be able to edit property details.
- As a Property Manager, I need to be able to edit my details.
- As a Property Manager, I need to be able to add units to a property.
- As a Property Manager, I need to be able to edt unit details.
- As a Property Manager, I need to be able to add a tenant to a unit.
- As a Property Manager, I need to be able to edit tenant details.
- As a Property Manager, I need to be able to add a sensor to a unit.
- As a Property Manager, I need to be able to edit sensor details.
- As a Property Manager, I need to be able to see the health of a sensor.
- As a Property Manager, I need to be able to view a unit's sound level history.
- As a Property Manager, I need to be able to assign rewards to a Unit.
- As a Property Manager, I need to be able to define acceptable decibel levels based on day of week and time of day.
- As a Property Manager, I need to be able to define the time period for earning a reward (e.g., days or months).
- As a Property Manager, I need to be able to update the status of a reward (e.g., claimed/unclaimed).
- As a Property Manager, I need to be able to generate summarized reports of noise violations.
- As a Property Manager I need to be able to open and update complaints.

User Role: Tenant

- As a Tenant, I need to be able to review my Unit’s sound level history.
- As a Tenant, I need to be able to see the health of my sensor.
- As a Tenant, I need to be able to initiate a complaint.
- As a Tenant, I need to be able to configure my details.
- As a Tenant, I need to be able to view my reward history.
- As a Tenant, I need to be able to receive real-time alerts when my unit's noise levels are too high.
- As a Tenant, I need to be able to configure notifications for my Unit’s sound level adherence.
- As a Tenant, I need to be able to add additional information to a noise complaint.